Security Flaws in Several Group Signatures Proposed by Popescu

Guilin Wang[1] and Sihan Qing[2]

[1] Infocomm Security Department (ICSD)
Institute for Infocomm Research (I^2R)
21 Heng Mui Keng Terrace, Singapore 119613
glwang@i2r.a-star.edu.sg
[2] ERCIST, Institute of Software
Chinese Academy of Sciences, Beijing 100080
qsihan@yahoo.com

Abstract. In resent years, Popescu proposed several group signature schemes based on the Okamoto-Shiraishi assumption in [8–11], and claimed his schemes are secure. However, this paper demonstrates that these schemes are all *insecure* by identifying some security flaws. Exploiting these flaws, an attacker without any secret can mount universally forging attacks. That is, anybody (not necessarily a group member) can forge valid group signatures on arbitrary messages of his/her choice.

Keywords: group signature, digital signature, information security.

1 Introduction

A group signature scheme, first introduced by Chaum and van Heyst in [6], allows each group member of a given group to sign messages anonymously on behalf of the group. However, in case of later disputes, a designated group manager can open a group signature and then identify the signer of it. Group signatures have many practical applications, such as e-voting, e-bidding, e-cash and fingerprinting systems, etc.

Following the first work by Chaum and van Heyst, a number of new group signature schemes and improvements have been proposed. Camenisch and Stadler proposed the first group signature scheme for large groups, since in their scheme the lengths of the group public key and signatures are independent of the group size [3]. In [4, 5], Camenisch and Michels constructed an efficient group signature scheme based on the strong RSA assumption. In 1999, Ateniese and Tsudik pointed out some obstacles that stand in the way of real world applications of group signatures, such as coalition attacks and member deletion [2]. Later, Ateniese et al. presented a practical and provably secure coalition-resistant group signature scheme [1]. To deal with exposure of group members' secret keys and deletion of group members, Song proposed forward-secure group signature schemes which support membership revocation [12].

Based on the Okamoto-Shiraishi assumption [7], Popescu recently proposed several group signature schemes. He first constructed two standard schemes in

[8, 9], and then extended them to a group blind signature [10] and a scheme with revocation [11]. Contrast to Song's schemes, Popescu's scheme in [11] has an advantage, that is, the system life time does not need to be divided into a predefined number of time periods. In the four schemes [8–11], the authors claimed that their schemes satisfy all the security requirements on group signatures (see Section 2 for details). However, this is not true.

In this paper, some serious security flaws in Popescu's schemes are successfully identified. Exploiting these flaws, an attacker can mount universally forging attacks without any secret. In other words, our attacks allow anybody (not necessarily a group member) to forge valid group signatures on arbitrary messages of his/her choice. This implies that these schemes are all insecure. Since these four schemes have similar structures, we only overview the most recent scheme proposed in [11] (For short, the PNBM scheme), and describe the related security flaws. Similar attacks also apply to other three schemes [8–10].

The rest of this paper is organized as follows. Section 2 first introduces the Okamoto-Shiraishi assumption [7], and the security requirements on a group signature scheme. We then review and analyze the PNBM scheme [11] in Section 3 and 4, respectively. Finally, some concluding remarks are given in Section 5.

2 Assumption and Security Requirements

In this section, we briefly review the Okamoto-Shiraishi assumption which all Popescu's schemes are based on, and the security requirements on group signatures.

Okamoto-Shiraishi Assumption [7] *Let e be an integer, $e \geq 4$. Given as inputs an RSA modulus $n = pq$ and an element $C \in \mathbb{Z}_n^*$, it is hard to find two integers X and δ such that $X^e \equiv C + \delta \bmod n$ and $\delta \in [a, b]$, where a and b are two integers satisfying* $0 \leq a < b < n^{2/3}$.

A secure group signature scheme must satisfy the following six properties [6, 3, 1, 2]:

1. **Unforgeability:** Only group members are able to sign messages on behalf of the group.
2. **Anonymity:** Given a valid signature of some message, identifying the actual signer is computationally hard for everyone but the group manager.
3. **Unlinkability:** Deciding whether two different valid signatures were computed by the same group member is computationally hard.
4. **Exculpability:** Neither a group member nor the group manager can sign on behalf of other group members.
5. **Traceability:** The group manager is always able to open a valid signature and identify the actual signer.
6. **Coalition-resistance:** A colluding subset of group members (even if comprised of the entire group) cannot generate a valid signature that the group manager cannot link to one of the colluding group members.

3 Review of the PNBM Group Signature Scheme

This section review the PNBM group signature scheme proposed by Popescu et al. in [11]. The whole scheme consists of six components.

3.1 SETUP

To setup a system, the group manager performs the following steps:

(1) Select two random safe primes p and q, i.e., there exist two primes p' and q' such that $p = 2p' + 1$ and $q = 2q' + 1$. Then, the group manager sets his RSA modulus $n = pq$. Let l_n denote the bit-length of n.
(2) Pick $\bar{G} = \langle \bar{g} \rangle$ of order n in which computing discrete logarithms is infeasible. For example, $\bar{G}$ can be a subgroup of $\mathbb{Z}_{\bar{p}}^*$ for a large prime $\bar{p}$ such that $n|(\bar{p}-1)$.
(3) Choose a public exponent e satisfying $e > 4$, and $\gcd(e, \varphi(n)) = 1$.
(4) Select an element g of order $2p'q'$ in $\mathbb{Z}_n^*$. Let $G = \langle g \rangle$, and l_G denote the bit-length of the order of G, i.e., $|G| = |\mathrm{ord}(g)| = l_G$ [1].
(5) Select an element $C \in \mathbb{Z}_n^*$ and an element $h \in_R G$ whose discrete logarithm to the base g must not be known.
(6) Pick a secret value $x \in_R \mathbb{Z}_n^*$ and computes $y = g^x \bmod n$.
(7) Publish a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^k$, and set security parameters $\epsilon > 1, l_1, l_2$.
(8) Finally, the public key is $PK = (n, e, g, \bar{g}, y, h, C, l_n, l_1, l_2, k, \epsilon, H)$ and the secret key is $SK = (p', q', x)$. In practice, components of PK must be verifiable to prevent framing attacks (refer to [4] for more detail).

An example for choosing the parameters is given by (see §5 of [9] or §2.1 of [11]): $l_n = 1200$, $l_1 = 350$, $l_2 = 240$, $k = 160$, $\epsilon = 5/4$, and $e = 5$.

3.2 JOIN

Suppose now that a user U_i wants to join the group. It is also assumed that the communications between the group member and the group manager is secure, i.e., private and authentic. A membership certificate in the PNBM group signature scheme consists of a pair of integers (X, δ) satisfying $X^e \equiv C + \delta \bmod n$ and $\delta \in [2^{l_1}, 2^{l_1} + 2^{l_2} - 1]$. To obtain his membership certificate, each user U_i must perform the following protocol with the group manager.

(1) The user U_i selects a random element $x_i \in [2^{l_1}, 2^{l_1} + 2^{l_2} - 1]$, and computes $ID_i = g^{x_i} \bmod n$.
(2) The user U_i must prove to the group manager that he knows $\log_g ID_i$ and that this value is in the interval $(2^{l_1} - 2^{\epsilon(l_2+k)+1}, 2^{l_1} + 2^{\epsilon(l_2+k)+1})$.

[1] In [11], this step is specified as follows: Select g an element of $\mathbb{Z}_n^*$ of order n. Let $G = \langle g \rangle$ be a cyclic subgroup of $\mathbb{Z}_n^*$ of order l_G. We note their specification is incorrect. Firstly, no element in $\mathbb{Z}_n^*$ has an order n, since $2p'q'$ is the maximum order of an element in $\mathbb{Z}_n^*$. Secondly, l_G should denote the bit-length of the order of G, not the order itself. So we correct these errors in our description.

(3) Then, the user U_i chooses a random number $r \in \mathbb{Z}_n^*$ and computes $z = r^e(C + x_i) \bmod n$. He sends z to the group manager.
(4) The group manager computes $v = z^{1/e} \bmod n = r(C + x_i)^{1/e} \bmod n$ and sends v to the user U_i.
(4) The user U_i computes $A_i = v/r = (C + x_i)^{1/e} \bmod n$. The pair (A_i, x_i) is the membership certificate of the user U_i.

Consequently, at the end of the protocol, the group manager does not know the membership certificate (A_i, x_i) of the user U_i. The group manager creates a new entry in the group database to store ID_i.

3.3 SIGN

With a membership certificate (A_i, x_i), a group member U_i can generate his group signature on any message $m \in \{0,1\}^*$ as follows.

(1) Select two random integers $w \in_R \{0,1\}^{l_2}$ and $r \in \mathbb{Z}_n^*$, and then compute: $A = A_i h^w \bmod n$, $B = g^w \bmod n$, $D = g^{x_i} y^w \bmod n$, $E = \bar{g}^r$, and $F = E^{b_s^{x_i}} \bmod n$. (Note that b_s is called the current revocation base, which is issued by the group manager in the `REVOKE` protocol. See the detail later.)
(2) Choose five random numbers $r_1 \in_R \{0,1\}^{\epsilon(l_2+k)}$, $r_2 \in_R \{0,1\}^{\epsilon(l_G+l_1+k)}$, $r_3 \in_R \{0,1\}^{\epsilon(l_G+k)}$, $r_4 \in_R \{0,1\}^{\epsilon(l_2+k)}$, $r_5 \in_R \{0,1\}^{\epsilon(l_2+k)}$, and then compute: $d_1 = B^{r_1}/g^{r_2} \bmod n$, $d_2 = g^{x_i^2} D^{r_4}/y^{r_5} \bmod n$, $d_3 = g^{r_3} \bmod n$, and $d_4 = g^{r_1} y^{r_3} \bmod n$.
(3) Evaluate the hash value $c = H(m||g||h||y||A||B||D||E||F||d_1||d_2||d_3||d_4)$.
(4) Calculate $s_1 = r_1 - c(x_i - 2^{l_1})$, $s_2 = r_2 - c x_i w$, $s_3 = r_3 - cw$, $s_4 = r_4 + x_i + c2^{l_1}$, $s_5 = r_5 + x_i w + c2^{l_1}$ (all in $\mathbb{Z}$).
(5) Release $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ as the group signature for message m.
(6) The user U_i proves in zero-knowledge that the double discrete logarithm of F with bases E and b_s, respectively, is the same as the discrete logarithm of D's representation to the bases g and h [2].

Since D is computed as $D = g^{x_i} y^w \bmod n$, the resulting proof of knowledge is verifiable if and only if the same x_i is used in the construction of both F and D.

3.4 VERIFY

Upon receiving an alleged group signature $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ on a message m, a verifier can check its validity as follows:

(1) Compute $d_1' = B^{s_1 - c2^{l_1}}/g^{s_2} \bmod n$, $d_2' = D^{s_4 - c2^{l_1}}/y^{s_5 - c2^{l_1}} \bmod n$, $d_3' = B^c g^{s_3} \bmod n$, and $d_4' = D^c g^{s_1 - c2^{l_1}} y^{s_3} \bmod n$.
(2) Evaluate the hash value $c' = H(m||g||h||y||A||B||D||E||F||d_1'||d_2'||d_3'||d_4')$.

[2] Note that there exist such protocols though not very efficient. For example, the one proposed in [3].

(3) Check whether $c \equiv c'$ and $s_1 \times s_2 \times s_3 \times s_4 \times s_5 \in \left\{-2^{l_2+k}, ..., 2^{\epsilon(l_2+k)}\right\} \times \left\{-2^{l_G+l_1+k}, ..., 2^{\epsilon(l_G+l_1+k)}\right\} \times \left\{-2^{l_G+k}, ..., 2^{\epsilon(l_G+k)}\right\} \times \left\{-2^{l_2+k}, ..., 2^{\epsilon(l_2+k)}\right\} \times \left\{-2^{l_2+k}, ..., 2^{\epsilon(l_2+k)}\right\}$.
(4) For each $V_{s,j} \in CRL$, check if $F \neq E^{V_{s,j}} \bmod n$.
(5) Check the proof of equality of double discrete logarithm fo F and the discrete logarithm of D's representation to the bases g and h.
(6) Accept the group signature $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ if only if all the above three checks hold.

3.5 `OPEN`

When a group signature $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ on a message m is given, the group manager can find out which one of the group members issued this signature by first checking its correctness via the `VERIFY` protocol. If the signature is not correct, he stops. Otherwise, the group manager performs the following steps to identify the signer.

(1) Recover $ID_i = D/B^x \bmod n$, and use the identity information ID_i to find the true signer U_i.
(2) Prove in zero-knowledge that $\log_g y = \log_B(D/ID_i \bmod n)$ [3, 9].

3.6 `REVOKE`

We begin by assuming, as usual, that a Certificate Revocation List (CRL) is a structure available at all time from a number of well-known public repositories or severs. A CRL is also assumed to be signed and timestamped by its issuer which may be a universally trusted CA, or the group manager. In addition, revocability means that a group signature produced using the `SIGN` algorithm by a revoked member must be rejected using the `VERIFY` algorithm.

We use s to denote the index of the current CRL issue where there are l group members to be revoked. The following `REVOKE` algorithm is executed by the group manager whenever a member or a collection of members leaves or is expelled.

(1) Choose a random number $b_s \in_R QR(n)$ of order $p'q'$. This value b_s becomes the *current revocation base*.
(2) For each U_j $(1 \leq j \leq l)$, compute $V_{s,j} = b_s^{x_j} \bmod n$.
(3) The actual revocation list is then published $CRL = \{b_s, V_{s,j} | 1 \leq j \leq l\}$.

4 Security Flaws in the PNBM Scheme

Popescu et al. claimed that their above scheme (and other schemes) satisfies all the security requirements listed in Section 2. However, this is not the fact.

4.1 REVOKE Algorithm

First of all, we note that the REVOKE algorithm proposed in the PNBM scheme does not work in the normal group signature framework.

The reason is that to issue each $V_{s,j}$ for all group members to be revoked, the group manager needs to know the value of x_i. However, x_i is U_i' member secret which cannot be revealed to anyone including the group manager. If this is not the case, using the value of x_i the group manager can recover the certificate A_i by computing $A_i = (C + x_i)^{1/e} \bmod n$. In this condition, the group manager can mount a *framing attack*, i.e., he can use the membership certificate (A_i, x_i) to generate valid signatures on behalf of the group member U_i. Of course, this is intolerable in the setting of group signatures since *exculpability* is not satisfied any more. This is a design error in their scheme.

According to the analysis presented as below, the PNBM scheme [11] (and the other three schemes) is not secure even in a setting where member revocation is not supported.

4.2 Security Parameters

Note that in the Camenisch-Michels scheme [4], the security parameters l_1 and l_2 are set as $l_1 = 860$, and $l_2 = 600$. But, in the PNBM scheme, the authors suggested to set the security parameters as $l_1 = 350$, and $l_2 = 240$ (see Section 2.1 of [11]). With much shorter exponents, the PNBM scheme may be more efficient [3]. However, the security parameters l_1 and l_2 should be selected as larger numbers. Especially, the difference between these two parameters should be guaranteed big enough. Otherwise, the schemes are vulnerable to some forging membership certificate attacks [13]. In general, the following condition is required [1]

$$l_1 > \epsilon(l_2 + k) + 2.$$

4.3 Cheating in the JOIN Protocol

In the third step of the JOIN protocol, user U_i is not required to prove that z and ID_i committed the same secret value of x_i. Therefore, a dishonest user U_i can replace x_i in z with a random number $\bar{x}_i \in_R [2^{l_1}, 2^{l_1} + 2^{l_2} - 1]$. That is, by choosing a random number $r \in_R \mathbb{Z}_n^*$, U_i prepares a value of $\bar{z}$ as

$$\bar{z} = r^e(C + \bar{x}_i) \bmod n,$$

and sends $\bar{z}$ to the group manager. Then, according to the JOIN protocol, the group manager sends back U_i the value of $\bar{v}$ which satisfies $\bar{v} = \bar{z}^{1/e} \bmod n$. Finally, U_i gets a valid membership certificates $(\bar{A}_i, \bar{x}_i)$ by computing

$$\bar{A}_i = \bar{v}/r \bmod n.$$

[3] We do not say that the PNBM scheme will become more efficient, since the signer has to execute a zero-knowledge protocol to show that he knows a double discrete logarithm. As we mentioned above, this is time-costing.

Using this valid certificate, U_i can generate valid group signatures normally. In later possible disputes, however, the group manager cannot open the signatures generated by such certificates. Because U_i uses $\bar{x}_i$ in stead of x_i in the `SIGN` protocol, and $\bar{x}_i$ has no any relationship with ID_i, the U_i's identity.

4.4 Universal Forgery

We note that in the `SIGN` protocol, the value of $A = A_i h^w \bmod n$ is not used in essence (except it is embedded in the hash value of c). In other words, what the `SIGN` protocol proves is that the signer knows some secrets such that the values of B and D are prepared properly. But whether A and D commits the same secret x_i is not proved. Therefore, anybody (not necessarily a group member) can generate a valid group signature for any message m of his choice as group member does.

To this end, an attacker first picks two random numbers $\bar{A}_i \in_R \mathbb{Z}_n^*$, and $\bar{x}_i \in_R [2^{l_1}, 2^{l_1} + 2^{l_2} - 1]$. Then, he can generate group signatures on any messages according to the procedures described in the `SIGN` protocol. It is easy to check that the resulting signatures are valid, i.e., they satisfies the `VERIFY` protocol.

5 Concluding Remarks

In this paper, we identified four security flaws in the group signature scheme with revocation by Popescu et al. [11]. Except the problem in the `REVOKE` algorithm, other security flaws can also be used to break the three schemes proposed in [8–10] by Popescu. The reason is that these group signature schemes have similar structure frameworks. Therefore, our results showed that all these schemes are completely *insecure*, and that the scheme in [11] does not support member revocation in essence. From our discussions presented above, we know that these security flaws mainly result from the insecurity of the `JOIN` and `SIGN` protocols, i.e., they are not designed securely. To improve these schemes, the `JOIN` and `SIGN` protocols should be carefully re-designed. Naturally, provably secure protocols are preferable and convincing. If the function of member deletion is necessary in the system, a new `REVOKE` algorithm has to be proposed, too.

References

1. G. Ateniese, J. Camenisch , M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In: *Crypto'2000, LNCS 1880*, pp. 255-270. Springer-Verlag, 2000.
2. G. Ateniese, and G. Tsudik. Some open issues and new directions in group signature schemes. In: *Financial Cryptography* (*FC'99*), *LNCS 1648*, pp. 196-211. Springer-Verlag, 1999.
3. J. Camenisch, and M. Stadler. Effient group signature schemes for large groups. In: *Crypto'97, LNCS 1294*, pp. 410-424. Springer-Verlag, 1997.
4. J. Camenisch and M. Michels. A group signature scheme with improved efficiency. In: *ASIACRYPT'98, LNCS 1514*, pp. 160-174. Springer-Verlag, 1998.

5. J. Camenisch and M. Michels. A group signature scheme based on an RSA-variant. *Technical Report RS-98-27*, BRICS, University of Aarhus, November 1998. An earlier version appears in [4].
6. D. Chaum and E. van Heyst. Group Signatures. In: *Eurocrypt'91, LNCS 950*, pp. 257-265. Springer-Verlag, 1992.
7. T. Okamoto and A. Shiraishi. A fast signature scheme based on quadratic inequalities. In: *Proceedings of IEEE Symposium on Security and Privacy*, pp. 123-132. IEEE Computer Society, 1985.
8. Constantin Popescu. Group signature schemes based on the difficulty of computation of approximate e-th roots. In: *Proc. of Protocols for Multimedia Systems* (*PROMS 2000*), pp. 325-331. Cracow, Poland, 2000.
9. Constantin Popescu. An efficient group signature scheme for large groups. *Studies in Informatics and Control Journal*, Vol. 10, No. 1: 7-14, March 2001. `http://www.ici.ro/ici/revista/sic2001-1/art1.htm`
10. Constantin Popescu. A group blind signature scheme based on the Okamoto-Shiraishi assumption. In: *Romanian Journal of Information Science and Technology*, Vol. 4, No. 5, 2002.
11. C. Popescu, D. Noje, B. Bede, and I. Mang. A group signature scheme with revocation. In: *Proc. of 4th EURSIP Conference focused on Video/Image Processing and Multimedia Communications* (*EC-VIP-MC 2003*), pp. 245-250. 2-5 July 2003, Zagreb, Coratia. IEEE Computer Society, available via IEEExplore `http://ieeexplore.ieee.org/`.
12. D.X. Song. Practical forward secure group signature schemes. In: *ACM CCS'01*, pp. 225-234. ACM press, 2001.
13. G. Wang, F. Bao, J. Zhou, and R.H. Deng. Security remarks on a group signature scheme with member deletion. In: *Information and Communications Security (ICICS 2003), LNCS 2836*, pp. 72-83. Springer-Verlag, 2003.